Onto-CARMEN: Ontology-driven approach for Cyber-Physical System Security Requirements meta-modelling and reasoning

Abstract

In the last years, Cyber-physical systems (CPS) have attracted substantial mainstream, especially in the industrial sector, since they have become the focus of cyber-attacks. CPS are complex systems that encompass a great variety of hardware and software components with a countless number of configurations and features. For this reason, the construction, validation, and diagnosis of security in CPS become a major challenge. An invalid security requirement for the CPS can produce partial or incomplete configuration, even misconfigurations, and hence catastrophic consequences. Therefore, it is crucial to ensure the validation of the security requirements specification from the earlier design stages. To this end, Onto-CARMEN is proposed, a semantic approach that enables the automatic verification and diagnosis of security requirements according to the ENISA and OWASP recommendations. Our approach provides a mechanism for the specification of security requirements on top of ontologies, and automatic diagnosis through semantic axioms and SPARQL rules. The approach has been validated using security requirements from a real case study.

Keywords: Cyber-Physical System, Cybersecurity, Security, Configuration Models, Security Requirements, Security Verification, Diagnosis

1. Introduction

Cyber-Physical Systems (CPS) are receiving mainstream attention within the industry, society and governments due to the enormous impact they have [1], and providing citizens and businesses with a wide range of innovative applications and services [2]. CPS are very complex systems composed of a great variety of components, sensors, controllers, computing, storage, protocols, and communication channels [3][4]. Their enormous capabilities allow for a wide range of new opportunities for industries and challenges such as those related to safety and security [5]. However, the design of a CPS (i.e., requirement engineering) is very challenging [6][7][8] since it involves elements from the real world, software layer, network layer, hardware layer, etc. In addition, the development of these systems has been carried out without thinking about the security aspects or the new risks that this automation of processes implies, which put the entire industrial infrastructure at risk [9]. Any security issue

could have catastrophic consequences [10]. Therefore, considering the security requirements from the earliest steps of CPS design is crucial to avoid security issues, even though it is challenging.

Security within industrial environments becomes a critical aspect [10] that must be assumed at all stages, from early analysis before the CPS is in place [11][8][12]. However, the enormous variability of the components involved in CPS and their possible configurations make it extremely difficult to define a correct bunch of security requirements [13][14]. Although security requirements are the appropriate solution in academia, for CPS, there are a paramount of physical, control and communication constraints and requirements, which make the task of defining security requirements and translating them into the design of the CPS even more complicated [15].

To manage this complexity, it is crucial to have a model that facilitates, on the one hand, tackling the variability of CPS components and their constraints and configurations, and security requirements and, on the other hand, enabling reasoning capabilities to ensure the correct design of security requirements. This problem has already been considered from the perspectives of variability [16] and semantic [17, 18]. However, both approaches focus on the security testing of critical systems based on the alignment of knowledge of threats and security requirements.

In previous work, we proposed CARMEN [19] as a framework to describe and diagnose security requirements. CARMEN is supported by two main elements, a meta-model for defining security requirements and a catalogue of variability models to gather CPS configuration constraints. CARMEN integrated weaving techniques to transform security requirement instances into configurations that are diagnosed through a specific variability model to explain why the security requirement is valid. CARMEN presents various drawbacks to the use of several models that are interrelated but unconnected, the dependency on ad-hoc weaving templates to connect the models (i.e., meta-model of security requirements and variability models), and the partial automation of the whole process of diagnosis.

In the context of security requirements, ontologies provide a shared understanding of the relevant concepts and their relationships, helping stakeholders to identify and articulate security needs more precisely. Ultimately, well-defined security requirements ontologies contribute to building more secure and resilient software systems.

Based on these limitations, Onto-CARMEN is proposed, a semantic approach to enable the modelling of security requirements for CPS and their instant verification and diagnosis at design-time thanks to the reasoning capabilities provided by ontologies. For this purpose, the main contributions of the Onto-CARMEN approach are:

1. The design and implementation of an ontology for CARMEN that enables the definition of security requirements for CPS according to the security recommendations of ENISA [3] and OWASP [20] guidelines.
2. A reasoning framework composed of semantic and SPARQL rules for the

verification and diagnosis of security requirements at design time that enable to derive the correct security requirements.

To explain in detail each of these steps, the paper is organised as follows: Section 2 introduces the background needed to understand the proposal. Section 3 describes the semantic model and rules to allow verification and diagnosis of security requirements for CPS. Section 4 applies Onto-CARMEN to different use cases to validate the approach. Section 5 reviews the most relevant papers; and, finally, conclusions are drawn and future work is outlined in Section 6.

2. Background

This section introduces the background required to understand our contributions. First, we present the main concepts related to the representation of ontologies. Next, we briefly describe CARMEN, a framework for specifying and diagnosing security requirements.

2.1. Ontology and representation of knowledge

An *ontology* is a formal model as an abstraction to represent real-world concepts [21]. It specifies the categories of entities in a domain and the properties of such entities.

There are common concepts related to the design and development of an ontology, such as classes (that represent categories that group entities with similar characteristics), individuals (specific instances or objects that belong to a particular class or category), attributes (properties of individuals), and relations (relationship between individuals).

Knowledge bases are constructed around an ontology. A knowledge base describes specific states in which a collection of unique instances of classes enables interoperability across different heterogeneous systems and databases [22].

There exist several alternatives of ontology languages [23], KIF, OWL, RDF+RDF(S) and DAML+OIL. The most popular language is *OWL*. OWL (Web Ontology Language) is a language designed to formalise ontologies whose expressiveness is greater than that proposed by RDF. OWL uses classes to define a group entity with similar characteristics, individuals as specific instances or objects that belong to a particular class or category, object properties as binary relationships between individuals, data properties as attributes of individuals, defined datatypes to represent allowed values for data properties, and axioms to represent cardinality and value constraints. The expressive resources of OWL include union, intersection, complement, and equivalence of concepts, functional, inverse, existential, and universal quantification in object properties and numerical restrictions. OWL2 is the next version of OWL and it incorporates expressive resources not included in OWL and distinguishes a set of fragments that are useful in practical applications. OWL2 is the latest W3C Recommendation for ontologies.

SPARQL [24] is a language designed to reasoning [25], query and modify ontologies. It is a language quite similar to SQL, the main difference being that

SQL assumes that the data are implemented in tables, and SPARQL assumes that the data are implemented in RDF graphs. Listing 1 shows an example of a query that modifies the ontology by deleting an element that fulfils a condition and inserting a new one.

Listing 1: Example of SPARQL rule

```
MODIFY
   DELETE {?x  <unidata#isTeachearin>  ?z}
   INSERT {?x  <unidata#isTeachearin>
               <unidata#English>}
   WHERE {?x  <unidata#isTeachearin>    ?z.
          FILTER(?z =  unidata#Spanish>)}
```

2.2. CARMEN framework

Figure 1 shows the steps of the CARMEN process. The process begins with the definition of security requirements for CPS. CARMEN contains a metamodel to support the definition of model instances as security requirements for CPS. Moreover, CARMEN uses feature models to represent the variability of configurations for CPS and security requirements. CARMEN maps security requirement instances with feature model concepts into security configurations through Model-Driven techniques such as weaving templates and transformations. Model weaving enables customising the model-to-model transformation according to the elements included in the requirement.

For this reason, the model-to-model transformation is fed (as shown in Figure 1) from the security requirement, the feature model, and the weaving templates. The result of the model weaving is a customised model-to-model transformation that enables the generation of a security configuration as an artefact to be verified and diagnosed. Thus, given a model instance as a security requirement for CPS, it is transformed through the model-to-model transformation (previously customised by the weaving) into security configurations. These configurations are diagnosed using automated analysis based on feature-oriented domain analysis engines [26]. The configurations are verified, but in case of invalid configurations, the diagnosis represents the explanation which turns an invalid configuration into a valid one.

The metamodel (see Figure 2) is focused on the security requirement entity that relates to a set of assets and security features. These assets have been aligned with the best security practises for IoT in the context of critical infrastructures formulated by the ENISA agency [3], and security features for CPS are obtained from OWASP [20] to extract the most important concepts (e.g., encryption, protocol, network, AES, SSL/TLS, Bluetooth, range, lifetime).

The asset comprises any physical, electronic, or virtual component (or set of them) presented in a CPS system. These components can be users, applications, services, platforms, devices, infrastructures or information. Besides, there may be relationships between individual components, such as between a device and the infrastructure that supports it.

The security feature represents the security concern for the requirements, e.g., the type of encryption for information or communication components. And

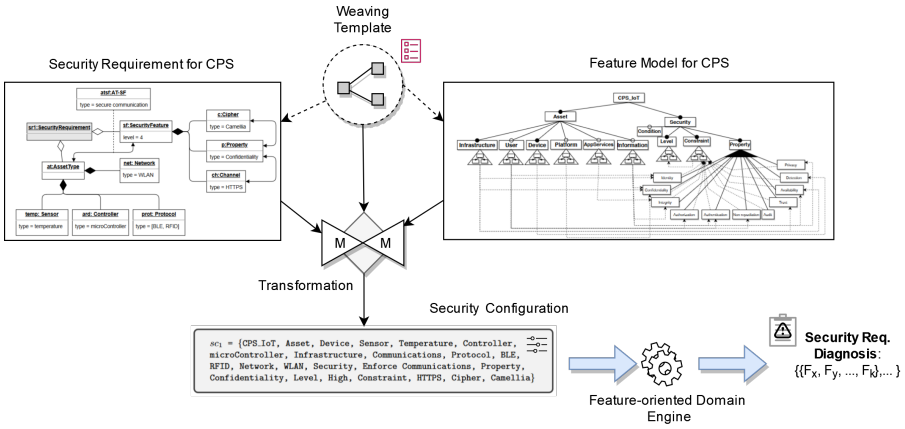

Figure 1: CARMEN process summary

the security property refers to Confidentiality or Integrity; a restriction must be defined concerning the type of encryption, or in the case of Authentication, a password policy must be defined.

Feature models [27] integrated with CARMEN as shown in Figure 3 are used as formal models to gather the variability of restrictions and relationships between assets and security features that must be considered for the configurations. For example, when confidentiality is required, enforce communications must be used as a secure channel. A feature model is a model which represents a set of products but is defined by their features and their relationships. A feature is a characteristic of the systems that can be configured, for instance, to choose the communication protocol from various alternatives. As can be seen, the feature model has encompassed two main parts: (1) the assets (cf., Asset) involved in the security requirement, and; (2) the security requirements (cf., Security) specification where properties, conditions, and constraints can be defined.

3. Onto-CARMEN approach

This section introduces the Onto-CARMEN approach. First, we introduce the process behind it, which consists of the definition of security requirements, their validation and diagnosis. Then, the semantic model is described in detail. Finally, the reasoning approach used to verify and diagnose is drawn.

3.1. Process overview

Figure 4 represents the workflow proposed by the Onto-CARMEN approach. Initially, it is necessary to describe the security requirements (cf., Define Security Requirement) that involve the CPS components and the security aspects. To this end, a semantic model for CPS security requirements is formalised in Section 3.2. The ontology permits the creation of individuals as new instances of security requirements.

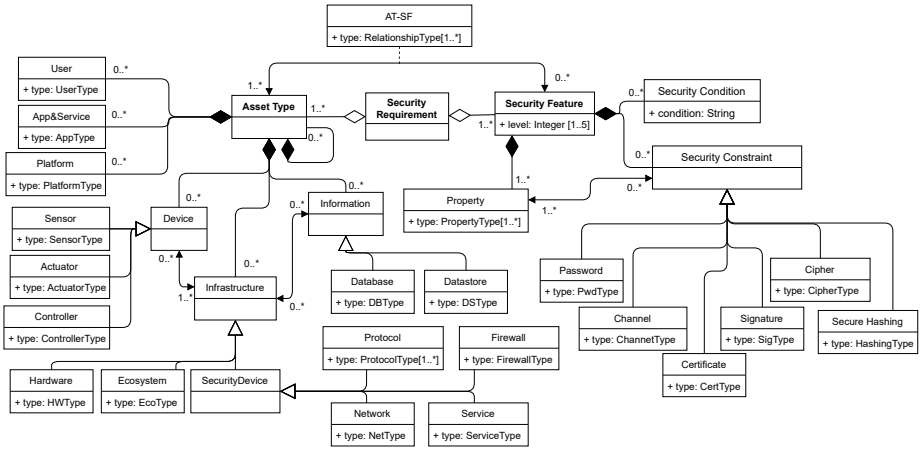

Figure 2: CARMEN metamodel

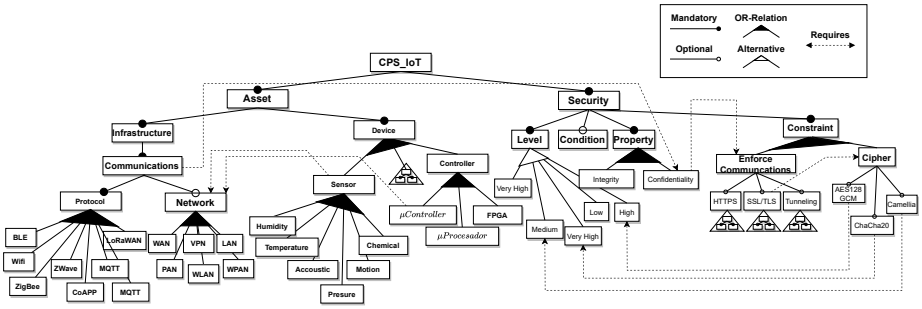

Figure 3: CARMEN features model

To validate these requirements (cf., Verify Security Requirements), the information gathered in the CARMEN feature models are translated into semantic axioms and SPARQL rules. Section 3.3 provides the necessary details to know the semantic rules and how they works. The validation through semantic rules enables a first checking of the security requirements identifying the validity of the requirement in (OK or KO). In the case of invalid security requirement, the semantic rules enable a diagnosis by providing corrective actions to transform into valid.

3.2. Semantic Model of CARMEN

This section focuses on Carmen's semantic model, first presenting the methodology used for its design and then presenting the resulting ontology describing its main elements.

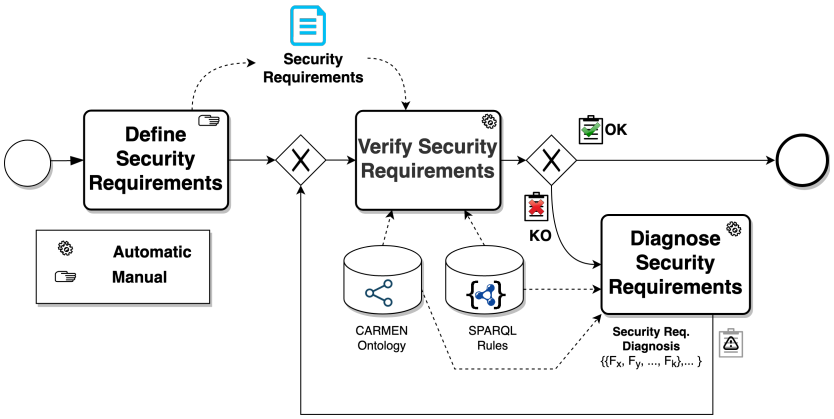

Figure 4: Onto-CARMEN process overview

3.2.1. Methodology

For the design of the ontology we have followed the well-known methodology "Ontology 101 development Process" [21]. The following provides an overview of how we have applied each step of the methodology.

- Determine the domain and scope of the ontology.

  The ontology focuses on the definition of security requirements for CPS. The requirements are associated with assets and their components (e.g. users, devices, infrastructure, etc.) and a set of security features necessary for their compliance (e.g. encryption, protocol, network, AES, SSL/TLS, Bluetooth, range, lifetime).

- Consider reusing existing ontologies.

  Considering the scope of our ontology, we need to handle concepts related to assets and security mechanisms applicable to the CPS domain and from the point of view of defining security requirements.

  We have not found ontologies that fully or partially cover the characteristics and relationships needed to represent our research scope and therefore we propose to develop an ontology from scratch. However, in the design of our ontology we have taken into account the reuse of the knowledge available in the community. We have reused standard concepts which, although they do not have specific ontologies, do define taxonomies of concepts. For the identification of assets and their components (e.g. users, devices, infrastructure, etc.) we have followed the ENISA [3] guidelines. For the associated security features we followed the OWASP [20] recommendations to extract the most important concepts (e.g. encryption, protocol, network, AES, SSL/TLS, Bluetooth, range, lifetime).

  We could extend specific concepts of this ontology by reusing other ontologies, e.g. by adding attack types, vulnerabilities, etc. However, with

the ontology developed we solve the problem in the scope of this work and those additional concepts would not be used neither in the definition of requirements nor in the reasoning about them for their diagnosis. However, in future work, we consider its evolution by extending it to support a more complete definition and reasoning on security requirements or to address other issues (e.g. threat intelligence knowledge, risk analysis and management, etc.). In these extensions, we will study whether there are ontologies that we can fully or partially integrate.

- Enumerate important terms in the ontology.

  The important ontology terms have been extracted from CARMEN's metamodel (Figure 2). These include SecurityRequirement, Asset, SecurityFeature, SecurityLevel, SustainabilityLabel, SecurityConstraint, etc.

- Define the classes and the class hierarchy.

  We extract ontology classes from relevant terms and define them along with their hierarchy to a sufficient level of detail for individual classification.

- Define the properties.

  We have established object and data properties that are necessary to represent the connections between individuals and to store their information.

- Define the facets of the slots.

  We have defined axioms for minimum and maximum cardinality constraints, possible values for the domain or range and other constraints such as transitivity, inverse, etc.

- Create instances.

  We have populated the ontology with specific individuals or instances that conform to the classes and properties defined in the ontology. These instances represent different security requirements and allow us to validate the ontology created.

3.2.2. Onto-CARMEN

This section presents the designed ontology, that consists of: 117 classes, 16 object properties, 2 data properties, 1 defined datatype, 12 individuals and 149 axioms. The following is a more detailed description of the main elements of the ontology.

- Classes

  Figure 5 displays the ontology classes and their relations. Only the main classes are shown in the figure and subclasses have been omitted, details of which are shown in Table 1. The main classes are shown below, together with a short description.

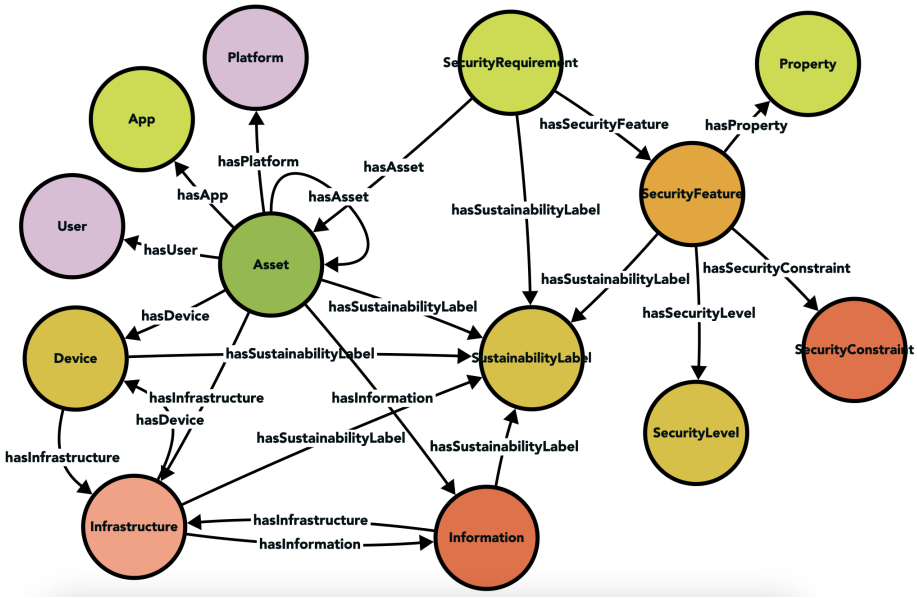

Figure 5: General overview of the ontology

Security Requirement. This class allows to specify the security needs and features that are defined in the CPS environment. For this purpose, both the set of assets involved in the definition of the requirement and the security features to be incorporated to adequately protect them should be indicated.

Asset. It is a term related to IT security risks that represents something that holds value within the system and needs to be protected. An asset can have one or more physical, electronic, or virtual components associated with it (in this case that we are talking about the CPS domain they can be users, applications, services, platforms, devices, infrastructures or information). It may be added to the requirement description in order to associate security features appropriate to the needs of the system to be protected.

Security Feature. It represents the security aspects identified for the requirements related to the assets. Thus, we can define what security restrictions are involved for the assets (e.g., the type of encryption for information or communication assets), which property or properties are defined in the requirement (authentication, authorization, confidentiality, and so on). A security level is also defined as varying from very low to very high.

Security level and sustainability label. They indicate security levels (very low, low, medium, high, very high) which can be associated with individuals of SecurityFeature and sustainability labels (A, B, C, D, E, F,

Table 1: Detail of the subclasses of Onto-CARMEN

App
AnalyticAndVisualizationApp, DeviceAndNetworkManagementApp, DeviceUsageApp
Asset
Device
Actuator
ElectricActuator, HydraulicActuator, MagneticActuator, MechanicalActuator, PnematicActuator, TCPSCPActuator, ThermalActuator
Controller
FPGAController, MicroController, MicroProcessorController
Sensor
AccousticSensor, ChemicalSensor, FlowmeterSensor, HumiditySensor, LuminositySensor, MotionSensor, PressureSensor, TemperatureSensor
Information
Database
GraphDBDatabase, NoSQLDatabase, SQLDatabase
Datastore
GPFSDatastore, HDFSDatastore, NFSDatastore
Infrastructure
Ecosystem
DeviceManageEcosystem, EmbeddedSystemsEcosystem, InterfaceEcosystem
Hardware
GatewayHardware, PowerSupplyHardware, RouterHardware
SecurityDevice
Firewall
HardwareFirewall, SoftwareFirewall
Network
LANNetwork, PANNetwork, VPNNetwork, WANNetwork, WLANNetwork, WPANNetwork
Protocol
BLEProtocol, CoAPPProtocol, LoRaWANProtocol, MQTTProtocol, RFIDProtocol, WifiProtocol, ZigBeeProtocol, ZWaveProtocol
Service
AuthenticationSystemService, CloudAuthenticationService, IDSIPSService
Platform
CloudInfrastructureAndServicesPlatform, WebBasedServicesPlatform
Property
AuditProperty, AuthenticationProperty, AuthorisationProperty, AvailabilityProperty, ConfidentialityProperty, DetectionProperty, IdentificationProperty, IntegrityProperty, NonRepudiationProperty, PrivacyProperty, TrustProperty
SecurityConstraint
Certificate
OpenPGPCertificate, OpenSSLCertificate, SAMLCertificate, X509Certificate
Channel
HTTPSChannel, SSLTLSChannel, TunnelingChannel
Cipher
AES128GCMCipher, CamelliaCipher, ChaCha20Cipher
Password
MultiFactorPassword, StrongPassword, WeakPassword
SecureHashing
SHA2SecureHashing, SHA3SecureHashing
Signature
PSKSignature, SRPSignature
SecurityFeature
SecurityLevel
SecurityRequirement
SustainabilityLabel
User
ConsumerUser, ProcessUser, ProviderUser, ThirdPartyUser

G) which can be associated with individuals of the majority of classes

(Infrastructure, Device, Platform, etc.).

- Properties

  Table 2 and 3 show in Description Logic format the set of defined object and data properties.

  Some examples of object properties are: "hasSecurityLevel" which indicates the security level associated with an instance of a SecurityFeature; "hasAsset" which indicates the set of particular assets involved in a Security Requirement; or "isLessSecureThan" and "isMoreSecureThan" that indicate relations between elements that have distinct security levels.

  On the other hand, several data properties have been defined. A security feature can have an associated expression to be evaluated as a condition. For this purpose we have defined the "condition" data property.

  Moreover, our proposal allows us to have several assets and each one of them to be linked with different, or the same, security feature (constraint, property or condition) with different types of relationship. For example, we can establish that the user (asset) access is controlled by a 2FA authentication, linking to the authentication and the password policy features, and has another asset such as wireless communications that are linked to the security features of confidentiality and some type of encryption. These relationships between assets and security features can have various types of relationships, such as secure communication and user authentication. This has been represented by using the data property "typeRelationship" which takes values from the datatype "TypeRelationship" (auditGuaranteed, authenticatedUser, authorisedUser, detectionSystem, encryptedInformation, identifiedUser, nonRepudiationGuaranteed, privacyGuaranteed, secureCommunication, trustGuaranteed).

- Facets of the slots

  Now, the axioms defined in the ontology are presented. We take as an example a concrete class (SecurityRequirement) to comment on the defined axioms. Table 4 shows the description logic corresponding to that class.

  The axioms related with SecurityRequirement class state that it has at least one asset, one relationship type and one security feature associated with it. It can also be associated (or not) to a sustainability label. In the securityRequirementHasSecurityFeature object property the domain is limited to individuals of the SecurityRequirement class, the range to individuals of the SecurityFeature class and the minimum cardinality is specified to be one (some).

  Some relationships are inverse such as hasSecurityConstraint and hasProperty. Others are transitive, such as the object properties isLess or isMore associated with security levels and sustainability labels in which one can be inferred to be greater or less than the other even though they do not have a direct connection.

Table 2: Object properties defined for Onto-CARMEN

Object property	Description logic
hasAsset	$\exists$ hasAsset Thing $\sqsubseteq$ Asset $\vee$ SecurityRequirement $\top \sqsubseteq \forall$ hasAsset Asset
hasInfrastructure	$\exists$ hasInfrastructure Thing $\sqsubseteq$ Device $\vee$ Information $\vee$ Asset $\top \sqsubseteq \forall$ hasInfrastructure Infrastructure
hasApp	$\exists$ hasApp Thing $\sqsubseteq$ Asset $\top \sqsubseteq \forall$ hasApp App
hasDevice	$\exists$ hasDevice Thing $\sqsubseteq$ Asset $\vee$ Infrastructure $\top \sqsubseteq \forall$ hasDevice Device
hasInformation	$\exists$ hasInformation Thing $\sqsubseteq$ Asset $\vee$ Infrastructure $\top \sqsubseteq \forall$ hasInformation Information
hasPlatform	$\exists$ hasPlatform Thing $\sqsubseteq$ Asset $\top \sqsubseteq \forall$ hasPlatform Platform
hasProperty	$\exists$ hasProperty Thing $\sqsubseteq$ SecurityFeature $\vee$ SecurityConstraint $\top \sqsubseteq \forall$ hasProperty Property
hasSecurityConstraint	$\exists$ hasSecurityConstraint Thing $\sqsubseteq$ SecurityFeature $\vee$ Property $\top \sqsubseteq \forall$ hasSecurityConstraint SecurityConstraint
hasSecurityFeature	$\exists$ hasSecurityFeature Thing $\sqsubseteq$ SecurityRequirement $\top \sqsubseteq \forall$ hasSecurityFeature SecurityFeature
hasSecurityLevel	$\exists$ hasSecurityLevel Thing $\sqsubseteq$ SecurityFeature $\top \sqsubseteq \forall$ hasSecurityLevel SecurityLevel
hasSustainabilityLabel	$\exists$ hasSustainabilityLabel Thing $\sqsubseteq$ App $\vee$ Asset $\vee$ Device $\vee$ Information $\vee$ Infrastructure $\vee$ Platform $\vee$ SecurityConstraint $\vee$ SecurityFeature $\vee$ SecurityRequirement $\vee$ User $\top \sqsubseteq \forall$ hasSustainabilityLabel SustainabilityLabel
hasUser	$\exists$ hasUser Thing $\sqsubseteq$ Asset $\top \sqsubseteq \forall$ hasUser User
isLessSecureThan	$\exists$ isLessSecureThan Thing $\sqsubseteq$ SecurityLevel $\top \sqsubseteq \forall$ isLessSecureThan SecurityLevel
isLessSustainabilityThan	$\exists$ isLessSustainabilityThan Thing $\sqsubseteq$ SustainabilityLabel $\top \sqsubseteq \forall$ isLessSustainabilityThan SustainabilityLabel
isMoreSecureThan	$\exists$ isMoreSecureThan Thing $\sqsubseteq$ SecurityLevel $\top \sqsubseteq \forall$ isMoreSecureThan SecurityLevel
isMoreSustainabilityThan	$\exists$ isMoreSustainabilityThan Thing $\sqsubseteq$ SustainabilityLabel $\top \sqsubseteq \forall$ isMoreSustainabilityThan SustainabilityLabel

Table 3: Data properties defined for Onto-CARMEN

Data property	Description logic
condition	$\exists$ condition Datatype Literal $\sqsubseteq$ SecurityFeature $\top \sqsubseteq \forall$ condition Datatype string
typeRelationship	$\exists$ typeRelationship Datatype Literal $\sqsubseteq$ SecurityRequirement $\top \sqsubseteq \forall$ typeRelationship Datatype RelationshipType

Table 4: Axioms for SecurityRequirement class

$\sqsubseteq \geq 1$ *hasAsset* . *Asset*
$\sqsubseteq \geq 1$ *typeRelationship* . *RelationshipType*
$\sqsubseteq \geq 1$ *hasSecurityFeature* . *SecurityFeature*
$\sqsubseteq (\geq 0$ *hasSustainabilityLabel* . *SustainabilityLabel*)
$\sqcap (\leq 1$ *hasSustainabilityLabel* . *SustainabilityLabel*)

On the other hand, value constraints have been used to restrict the values of the data property "typeRelationship" to a set of values defined in an own datatype "RelationshipType" (auditGuaranteed, authenticatedUser, authorisedUser, detectionSystem, encryptedInformation, identi-

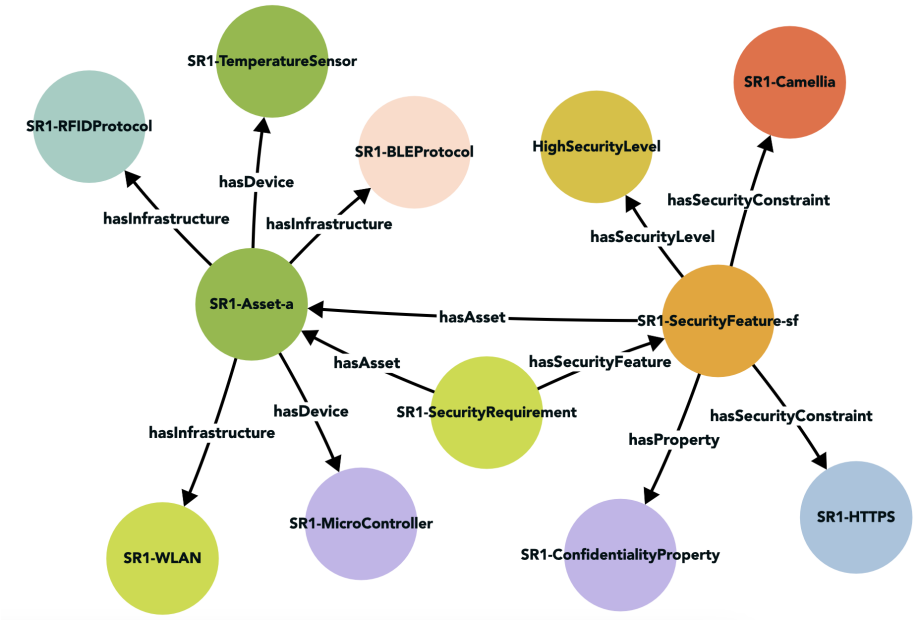

Figure 6: Security requirement SR1 represented with Onto-CARMEN

fiedUser, nonRepudiationGuaranteed, privacyGuaranteed, secureCommunication, trustGuaranteed).

- Instances

  For the validation of the ontology several instances have been created such as the one shown in Figure 6 in which a Security Requirement (SR1) is represented with an asset (composed of network, protocol, sensor, controller) and a security feature (composed of channel, property, cipher) associated with it. The figure shows the object properties and the data property values are detailed in Table 5.

  The requirement states that the wireless communication between the temperature sensor and the Arduino micro-controller must be encrypted in order to maintain a high level of confidentiality. This requirement also establishes a security feature that specifies that the communication must be encrypted in accordance with confidentiality standards. To achieve this, we have defined the property of Confidentiality, which is associated with a specific type of encryption (Camellia) and a secure communication channel (HTTPS). This security feature applies to an asset that includes the temperature sensor and Arduino, which communicate with each other over a WLAN network using either BLE or RFID. This relationship between the security feature and the asset is referred to as “secure communication”.

Table 5: Detail of the data properties associated to SR1

Instance	Class	Data Property	Value
SR1-SecurityRequirement	SecurityRequirement	typeRelationship	secureCommunication

3.3. Reasoning framework

As previously explained in Section 2, there is a set of feature models that gathers the restrictions and relationships between elements that must be taken into account for the security configurations created to be valid. In Onto-CARMEN these constraints have been defined by means of a set of SPARQL rules instead of in the OWL ontology itself. This decision allows us not only to reason about the ontology for the validation of security configurations, but also to diagnose and repair them in an automated way. This means that each defined rule checks a certain invalid situation and, in case of detecting it, makes the necessary changes (modifying properties, adding or deleting instances, etc.) to convert it into a valid configuration.

To represent all the necessary constraints, we have defined a SPARQL rule for each constraint present in the CARMEN feature models. The rule set is applied iteratively until the security requirement is valid. That is to say, a series of rules will be triggered by a situation that may remove or add elements, thus constituting another situation that may be valid or that may trigger other rules. In this way, valid configurations are achieved in an iterative and incremental manner.

Several examples are presented below. In the first one we focus on a restriction that indicates that if in a security feature, we are using a Camellia cipher, the security level associated with that security feature has to be medium. The rule CamelliaRequiresMediumSecurityLevel shown below is in charge of detecting situations in which this type of cipher is being used but the security level is not adequate and, if found, it would change the security level associated with the security feature to the correct one (medium in this case).

Figure 7 shows how the rule is applied to the security requirement SR1 (previously presented), detecting an invalid situation (marked in red) since it presents a high security level and how after applying the rule, the security level has been modified to medium (marked in green).

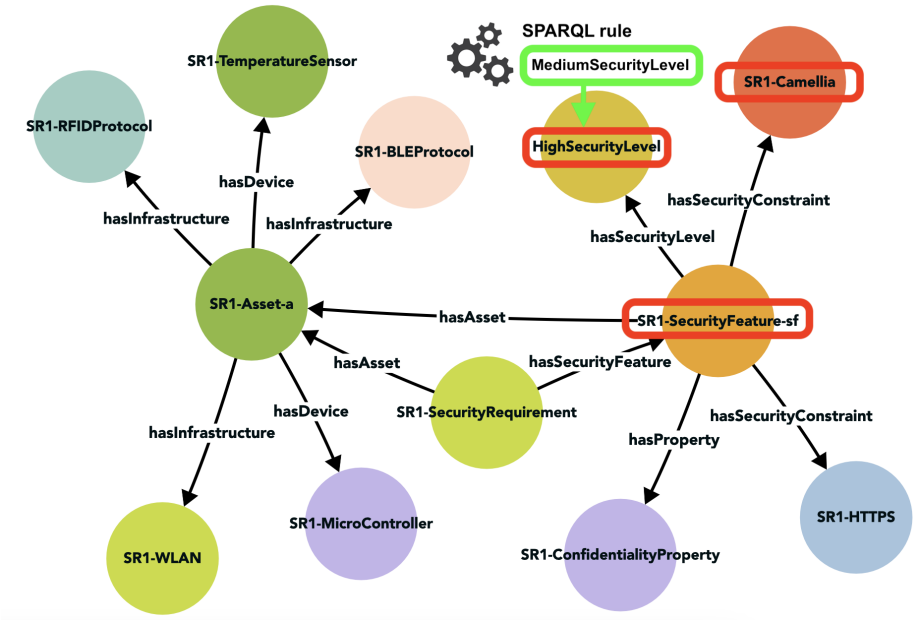

Figure 7: Application of the CamelliaRequiresMediumSecurityLevel rule to SR1

CamelliaRequiresMediumSecurityLevel SPARQL rule

```
DELETE {
    ?sf ontocarmen:hasSecurityLevel ?sl .
}

INSERT {
    ?sf ontocarmen:hasSecurityLevel
    ontocarmen:MediumSecurityLevel .
}

WHERE{
   ?sf a ontocarmen:SecurityFeature .
   ?sf ontocarmen:hasSecurityLevel ?sl .
   FILTER (?sl != ontocarmen:MediumSecurityLevel) .
   ?sf ontocarmen:hasSecurityConstraint ?sc .
   ?sc a ontocarmen:CamelliaCipher .
}
```

In the following example, we focus on a restriction that indicates that if there is a security feature that requires confidentiality it must enforce communications by using a secure channel. Next we show the rule ConfidentialityRequiresChannel that is in charge of capturing the non-compliance of this restriction and of solving it by creating a channel and associating it to the security feature in an appropriate way.

ConfidentialityRequiresChannel SPARQL rule

```
INSERT {
    ?ch a ontocarmen:Channel .
    ?sf ontocarmen:hasSecurityConstraint ?ch
}

WHERE {
    ?sf a ontocarmen:SecurityFeature .
    ?sf ontocarmen:hasProperty ?p .
    ?p a ontocarmen:ConfidentialityProperty.

    FILTER NOT EXISTS {
        ?sf ontocarmen:securityFeatureHasSecurityConstraint ?sc .
        ?sc a ontocarmen:Channel . }
}
```

4. Experimental validation

In this section, various security requirements from a real-case study are employed to validate the approach in terms of reasoning capabilities for validation and verification of security requirements.

The security requirements are obtained from a CPS system for hydroponic farming [19], in which various components are involved, both hardware and software. Hydroponic farming is controlled by the following physical elements: temperature, light and humidity sensors; heater, cooler, ultraviolet, nutrient injector and water pump actuators; a controller that is an Arduino device that receives the data from all the sensors and sends it (via wireless connections) through a web system to a Big Data system for subsequent storage, analysis and display. In addition to the physical part, the controller is connected to a visualisation and control system with Big Data technologies where a dashboard is deployed, along with a data handler, and a datastore (HDFS and HBASE).

The security requirements of the case study are described below.

- **Security requirement SR1:** This requirement, that defines an enforcement of communications between sensors and controllers of the CPS, has been described in previous section.

- **Security requirement SR2:** The short-range sensors of temperature, light and humidity are connected to an Arduino controller via Bluetooth. The transmitted information acts under the HTTP client/server protocol but the transmitted information must be secured by applying the SSL/TLS protocol over HTTP, ensuring confidentiality. This information is stored encrypted in a local webserver with HDFS and HBASE, ensuring integrity. Figure 8 shows the SR2 requirement but omitting some object properties to make it clearer and showing the data properties in Table 6.

  First, SR2 defines that the communication must be confidential with a high level, so the security feature is defined which has the property of

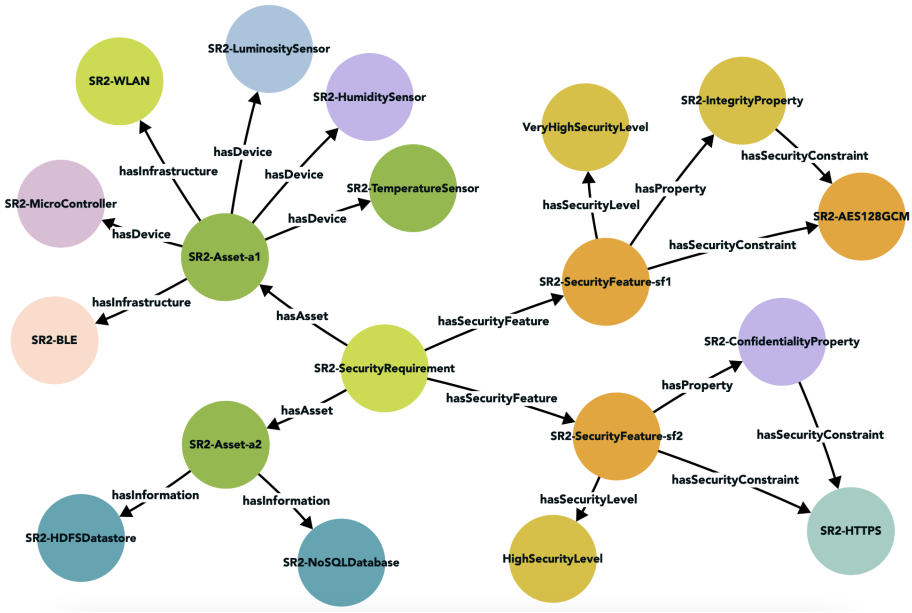

Figure 8: Security requirement SR2 represented with Onto-CARMEN

Table 6: Detail of the data properties associated to SR2

Instance	Class	Data Property	Value
SR2-SecurityRequirement	SecurityRequirement	typeRelationship	secureCommunication
			encryptedInformation

confidentiality and a secure communication channel using HTTPS. These conditions are checked using the SPARQL rules shown in Section 3.3. However, this is invalid: the database and datastore are being related to a security feature with a *very* high-security level using AES128GCM encryption, whereas this cipher is just indicated for high-security level. The next SPARQL rule enables to detect this invalid situation.

AES128GCMRequiresHighSecurityLevel SPARQL rule

```
DELETE {
  ?sf ontocarmen:hasSecurityLevel ?sl .
}

INSERT {
  ?sf ontocarmen:hasSecurityLevel
  ontocarmen:HighSecurityLevel .
}

WHERE{
   ?sf a ontocarmen:SecurityFeature .
   ?sf ontocarmen:hasSecurityLevel ?sl .
   FILTER (?sl != ontocarmen:HighSecurityLevel) .
   ?sf ontocarmen:hasSecurityConstraint ?sc .
   ?sc a ontocarmen:AES128GCMCipher .
}
```

After diagnosing SR2, there are two options as corrective actions: (1) lowering the security level from very high to high to comply with the conditions of AES128GCM; or (2) changing the encryption type to ChaCha20. In our proposal, we have defined a SPARQL rule for each constraint, so the rule implements one of the possible solutions (in this case, the defined rule would change the security level from very high to high). As an improvement point of our proposal, additional rules could be defined for each possible solution. These rules would check more elements related to that situation, and the rule for the most appropriate solution would be triggered. However, this approach would increase the number of rules in our proposal very considerably.

- **Security requirement SR3:** The nutrient injector is a critical device for the hydroponic farming and therefore integrity, confidentiality and authentication must be guaranteed in all actions performed with it. Therefore, only a user with a valid x509 digital certificate will be in charge of sending orders to the device and this access will be registered. Figure 9 shows the SR3 requirement omitting some object properties that are detailed in Table 7.

Table 7: Detail of the data properties associated to SR3

Instance	Class	Data Property	Value
SR3-SecurityRequirement	SecurityRequirement	typeRelationship	auditGuaranteed, authenticatedUser, secureCommunication

Similar to SR2, SR3 defines that communication must be confidential at a high level, so the security feature is defined which has the property of confidentiality and a secure communication channel using HTTPS. On the other hand, SR3 defines that users must access through an authentication

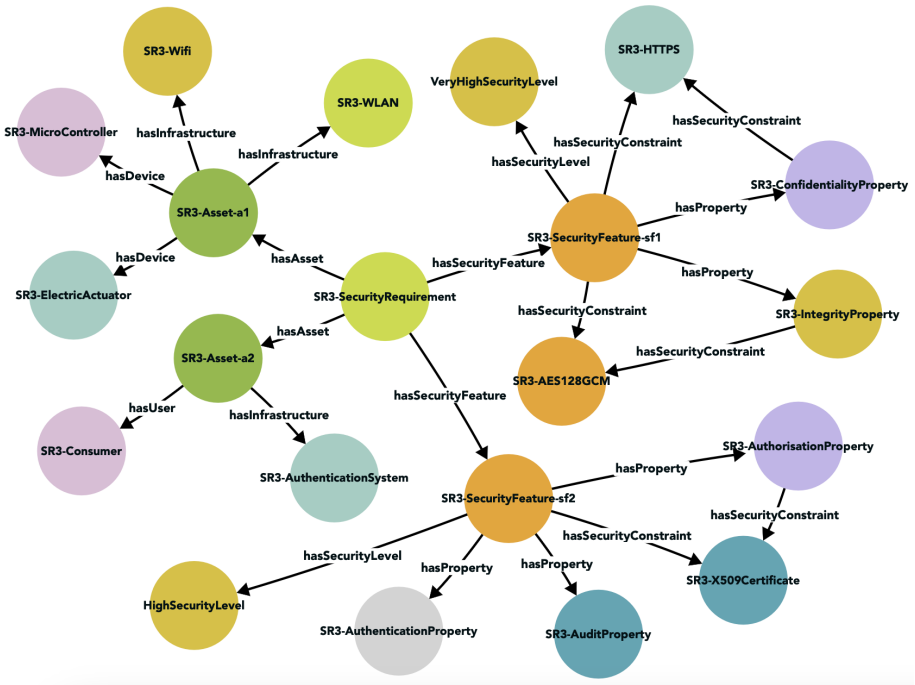

Figure 9: Security requirement SR3 represented with Onto-CARMEN

service with a high security level, compliance with authentication and authorisation through a user's x509 certificate and must comply with correct user status as a condition for access. Further, auditing property must be guaranteed through the registration of all the accesses of any assets made by authentication service.

SR3 is an invalid requirement due to two main conditions: (1) the encryption is invalid for high security level; and, (2) the use of authentication services requires the enforcement that is not established in the requirement.

AuthenticationAndHighSLRequiresMultifactorPassword SPARQL rule

```
INSERT {
    ?ch pa :Password .
    ?pa :typePassword "multiFactor" .
    ?sf :securityFeatureHasSecurityConstraint ?ch
}

WHERE {
    ?sf a :SecurityFeature .
    ?sf :securityFeatureHasProperty ?p .
    ?p :typeProperty "authentication" .
    ?sf :securityFeatureHasSecurityLevel ?sl .

    FILTER (?sl = ontocarmen:HighSecurityLevel) .

    FILTER NOT EXISTS {
        ?sf :securityFeatureHasSecurityConstraint ?sc .
        ?sc a :Password }
}
```

After diagnosing SR3, the proposed SPARQL rules enables various options as corrective actions: (1) in the same way as SR2, lowering the security level from very high to high to comply with the conditions of AES128GCM; and (2) including the enforcement of authentication using a password policy mechanism according to the high security level, thus, a Multifactor password system.

5. Related Work

In this section, we survey some of the existing ontologies related to specific security requirements, particularly for CPS. We will also study the ontologies of security requirements for analysis and risk management.

There are numerous works dedicated to the study of security ontologies from different phases and purposes of software engineering [12, 17, 28, 29]. From the analysis and classification of security requirements [30, 31, 32] to the design and use of patterns [33, 34] and tools [35, 36].

There are other works that focus their studies on security ontologies in specific fields such as Cloud computing [37, 38] and IoT [39, 40, 41]. There

are also works applied to specific environments or scenarios such as healthcare [42, 43, 44, 45] or intelligent systems [46, 47, 48], to mention just a few.

On the other hand, the security ontologies for risk analysis and management allow to obtain automatically an overview of the whole system. These ontologies are used to infer the necessary knowledge and estimate the risk level of the analysed system [49]. There are some works on the definition and use of security ontologies for risk analysis and management [50, 51, 52, 53]. Other works are related to the cybersecurity vulnerability ontology or attacks [54, 55, 56]. In addition, in [57], the authors perform an ontological analysis of the impact of defects, errors and failures of software systems from a risk analysis perspective.

As we can see, security ontologies is a well-studied topic where we can find numerous proposals. Most of them are focused on how to define, design and use them to model and classify, for example, security requirements, which is what we propose in this paper, but oriented to a CPS environment. If we focus on this type of environment, there are few proposals found that can be related to the definition and modelling of security requirements. We can highlight the work by Shaaban et. al. [18] who introduce an ontology-based security tool-chain able to be integrated with the initial stages of the development process of critical systems. The tool detects the potential threats, and applies the suitable security requirements which can address these threats and ensure that the security requirements are fulfilled. Our proposal, for the time being, and unlike the work of Shaaban et. al., does not take threats into account to establish security requirements, i.e., it is not threat-driven. Our proposal focuses on the correct and consistent definition of a security requirement, and its subsequent validation, with all possible characteristics of CPS environments, i.e., it is driven by security requirements. In [58] authors present an ontology-driven framework that captures the relationship between cyber and physical systems to semantically reason about the impact of cyber-attacks on the physical systems. The objective of this proposal is to reason about possible attacks in a specific scenario, so it does not offer any mechanism to define security requirements that could protect the system from such possible attacks from the beginning of the development as we offer in our proposal. Other work related to CPS is presented by Balduccini et. al. [59] who take the model created by the CPS framework by applying ontological approaches and reasoning techniques in order to achieve a greater understanding of CPS. This work formally models the trustworthiness aspect to check whether the impacts that occur satisfy the cibersecurity properties and what actions need to be taken. It does not model any security requirements or possible security configurations in these environments. Our proposal provides the necessary mechanisms to define the security requirements or needs of a CPS and to diagnose whether they are well defined and satisfy the properties of the CPS environments.

As we have previously indicated, some proposals provide security ontologies to define, model, and classify security requirements. Many proposals focus on system development or specific contexts related to CPS such as IoT systems. These proposals focus more on policies, communications, or specific domains. Our proposal focuses on a specific environment such as CPS systems where any

domain that makes use of CPS concepts has a place. In addition, our proposal focuses on helping in the early stages of development by identifying, modelling, and analyzing security requirements. This allows to conduct a verification and validation of the requirements, finding incompatibilities in their security configuration or poor definition and inconsistency during the security requirements elicitation phase.

6. Conclusions and future work

In this paper, we propose the design and implementation of a security ontology called Onto-CARMEN. Onto-CARMEN is capable of capturing the security needs of CPS systems through the definition of security requirements. The elements that will be part of the requirements are identified following the security recommendations of the ENISA and OWASP guides. In addition, once Onto-CARMEN has been created, a reasoning framework is proposed incorporating the necessary SPARQL queries for the verification and diagnosis of security requirements at design time. This allows users to know at all times whether or not the definition of the requirement is adequate and complies with the ontological rules. It also allows users to detect incorrect security requirements definitions and derive or recommend correct security requirements according to the defined ontology.

The main limitations of CARMEN were the use of different unconnected models, the dependency on ad-hoc weaving templates to connect the models, and the partial automation of the whole process of diagnosis. Thus, CARMEN needs to transform a security requirement into an intermediate artefact (configuration) to diagnose it. However, Onto-CARMEN presents an automatic process for the verification and diagnosis of security requirements for CPS at design time. That is, Onto-CARMEN can ensure that the security requirement is according to CPS and the ENISA and OWASP guidelines just by using the ontology and a set of rules. Although the Onto-CARMEN ontology only collects the ENISA and OWASP recommendations, the use of ontologies presents great advantages, such as interrelating concepts with other ontologies from different domains. Finally, it is important to clarify that Onto-CARMEN is not devised to analyse the entire design of a CPS but to security requirements for CPS.

In future work, we will incorporate other existing security ontologies to extend and enrich Onto-CARMEN by incorporating threat intelligence knowledge and risk analysis and management ontologies. We also plan to extend our proposal to add the knowledge and reasoning engine to predict attacks and risks as soon as a security requirement is defined.

For future work we will use the NeON methodology [60], which stands out for its focus on the reuse of resources in the creation of ontologies. In addition, we propose the use of SHACL (Shapes Constraint Language) for the definition of rules in a way that preserves the declarative nature of ontologies.

In addition, we intend to make system security configurations more sustainable. For this purpose, we intend to incorporate concepts of sustainability and

energy efficiency and consumption for all types of assets and all security features of CPS systems. This will add to our proposal the ability to predict and recommend the best and most efficient security mechanisms and configurations for a specific CPS environment.

Acknowledgement

Omitted by blind review.

References

[1] O. Mörth, C. Emmanouilidis, N. Hafner, M. Schadler, Cyber-physical systems for performance monitoring in production intralogistics, Computers & Industrial Engineering 142 (2020) 106333. doi:https://doi.org/10.1016/j.cie.2020.106333.

[2] A. W. Colombo, G. J. Veltink, J. Roa, M. L. Caliusco, Learning industrial cyber-physical systems and industry 4.0-compliant solutions, in: 2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS), Vol. 1, IEEE, 2020, pp. 384–390.

[3] Baseline security recommendations for IoT (Apr 2018).
URL https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot

[4] Y. Maleh, Machine learning techniques for iot intrusions detection in aerospace cyber-physical systems, in: Machine Learning and Data Mining in Aerospace Technology, Springer, 2020, pp. 205–232.

[5] H. Mokalled, C. Pragliola, D. Debertol, E. Meda, R. Zunino, A comprehensive framework for the security risk management of cyber-physical systems, in: Resilience of Cyber-Physical Systems, Springer, 2019, pp. 49–68.

[6] J. Geismann, C. Gerking, E. Bodden, Towards ensuring security by design in cyber-physical systems engineering processes, in: Proceedings of the 2018 International Conference on Software and System Process, ICSSP '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 123–127. doi:10.1145/3202710.3203159.
URL https://doi.org/10.1145/3202710.3203159

[7] S. Peisert, J. Margulies, D. M. Nicol, H. Khurana, C. Sawall, Designed-in security for cyber-physical systems, IEEE Security Privacy 12 (5) (2014) 9–12. doi:10.1109/MSP.2014.90.

[8] S. ur Rehman, C. Allgaier, V. Gruhn, Security requirements engineering: A framework for cyber-physical systems, in: 2018 International Conference on Frontiers of Information Technology (FIT), IEEE, 2018, pp. 315–320.

[9] M. Lezzi, M. Lazoi, A. Corallo, Cybersecurity for industry 4.0 in the current literature: A reference framework, Computers in Industry 103 (2018) 97 – 110. doi:https://doi.org/10.1016/j.compind.2018.09.004.

[10] Cyber-physical systems security: Limitations, issues and future trends, Microprocessors and Microsystems 77 (2020) 103201. doi:https://doi.org/10.1016/j.micpro.2020.103201.

[11] C. Zunino, A. Valenzano, R. Obermaisser, S. Petersen, Factory communications at the dawn of the fourth industrial revolution, Computer Standards & Interfaces 71 (2020) 103433. doi:https://doi.org/10.1016/j.csi.2020.103433.

[12] A. Souag, C. Salinesi, R. Mazo, I. Comyn-Wattiau, A security ontology for security requirements elicitation, in: International symposium on engineering secure software and systems, Springer, 2015, pp. 157–177.

[13] P. Valle, A. Arrieta, M. Arratibel, Automated misconfiguration repair of configurable cyber-physical systems with search: an industrial case study on elevator dispatching algorithms, CoRR abs/2301.01487. arXiv:2301.01487, doi:10.48550/arXiv.2301.01487.
URL https://doi.org/10.48550/arXiv.2301.01487

[14] Á. J. Varela-Vaca, D. G. Rosado, L. E. Sánchez, M. T. Gómez-López, R. M. Gasca, E. Fernández-Medina, Definition and verification of security configurations of cyber-physical systems, in: S. Katsikas, F. Cuppens, N. Cuppens, C. Lambrinoudakis, C. Kalloniatis, J. Mylopoulos, A. Antón, S. Gritzalis, W. Meng, S. Furnell (Eds.), Computer Security, Springer International Publishing, Cham, 2020, pp. 135–155.

[15] B.-J. Kim, S.-W. Lee, Understanding and recommending security requirements from problem domain ontology: A cognitive three-layered approach, Journal of Systems and Software 169 (2020) 110695. doi:https://doi.org/10.1016/j.jss.2020.110695.

[16] A. Arrieta, G. Sagardui, L. Etxeberria, Cyber-physical systems product lines: Variability analysis and challenges, 2015.

[17] M. Span, L. O. Mailloux, R. F. Mills, W. Young, Conceptual systems security requirements analysis: Aerial refueling case study, IEEE Access 6 (2018) 46668–46682.

[18] A. M. Shaaban, T. Gruber, C. Schmittner, Ontology-based security tool for critical cyber-physical systems, in: 23rd International Systems and Software Product Line Conference - Volume B, SPLC '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 207–210. doi:10.1145/3307630.3342397.

[19] Á. J. Varela-Vaca, D. G. Rosado, L. E. Sánchez, M. T. Gómez-López, R. M. Gasca, E. Fernández-Medina, CARMEN: A framework for the verification and diagnosis of the specification of security requirements in cyber-physical systems, Computers in Industry 132 (2021) 103524. doi:10.1016/j.compind.2021.103524.

[20] OWASP Internet of Things Project, Available from OWASP (2021). URL https://wiki.owasp.org/index.php/OWASP_Internet_of_Things_Project#tab=Main

[21] N. Noy, D. M. (Hrsg.), Ontology development 101: A guide to creating your first ontology. Technical report, Stanford knowledge systems laboratory technical report KSL-01-05 and ..., 2001.

[22] W. Yun, X. Zhang, Z. Li, H. Liu, M. Han, Knowledge modeling: A survey of processes and techniques, International Journal of Intelligent Systems 36 (4) (2021) 1686–1720.

[23] D. Kalibatiene, O. Vasilecas, Survey on ontology languages, in: J. Grabis, M. Kirikova (Eds.), Perspectives in Business Informatics Research, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 124–141.

[24] S. Harris, A. Seaborne, SPARQL 1.1 Query Language (Mar. 2013). URL https://www.w3.org/TR/2013/REC-sparql11-query-20130321/

[25] D. Anicic, P. Fodor, S. Rudolph, N. Stojanovic, Ep-sparql: a unified language for event processing and stream reasoning, in: Proceedings of the 20th international conference on World wide web, 2011, pp. 635–644.

[26] D. Benavides, S. Segura, A. Ruiz-Cortés, Automated analysis of feature models 20 years later: A literature review, Information Systems 35 (6) (2010) 615 – 636. doi:https://doi.org/10.1016/j.is.2010.01.001.

[27] Á. J. Varela-Vaca, R. M. Gasca, Formalization of security patterns as a means to infer security controls in business processes, Logic Journal of the IGPL 23 (1) (2015) 57–72. doi:10.1093/jigpal/jzu042.

[28] A. M. Shaaban, T. Gruber, C. Schmittner, Ontology-based security tool for critical cyber-physical systems, in: 23rd International Systems and Software Product Line Conference - Volume B, SPLC '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 207–210. doi:10.1145/3307630.3342397.

[29] A. Rashid, R. Ali, W. Lages, Ontology-based security requirements engineering for software-intensive systems, IEEE Transactions on Software Engineering 45 (2) (2019) 187–215. doi:10.1109/TSE.2017.2763965.

[30] H. Alrumaih, A. Mirza, H. Alsalamah, Domain ontology for requirements classification in requirements engineering context, IEEE Access 8 (2020) 89899–89908. doi:10.1109/ACCESS.2020.2993838.

[31] T. Li, Z. Chen, An ontology-based learning approach for automatically classifying security requirements, Journal of Systems and Software 165 (2020) 110566. doi:https://doi.org/10.1016/j.jss.2020.110566.
URL https://www.sciencedirect.com/science/article/pii/S0164121220300479

[32] R. Guizzardi, G. Amaral, G. Guizzardi, J. Mylopoulos, Eliciting ethicality requirements using the ontology-based requirements engineering method, Lecture Notes in Business Information Processing 450 (2022) 221 – 236, cited by: 0; All Open Access, Green Open Access. doi:10.1007/978-3-031-07475-2_15.
URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85131314326&doi=10.1007%2f978-3-031-07475-2_15&partnerID=40&md5=13754867fce024874a7851149b56830a

[33] A. P. Vale, E. B. Fernandez, An Ontology for Security Patterns, Proceedings - International Conference of the Chilean Computer Science Society, SCCC 2019-November. doi:10.1109/SCCC49216.2019.8966393.

[34] H. Guan, H. Yang, J. Wang, An ontology-based approach to security pattern selection, International Journal of Automation and Computing 13 (2) (2016) 168–182. doi:10.1007/s11633-016-0950-1.
URL http://link.springer.com/10.1007/s11633-016-0950-1

[35] S. Peldszus, J. Bürger, T. Kehrer, J. Jürjens, Ontology-driven evolution of software security, Data & Knowledge Engineering 134 (2021) 101907. doi:https://doi.org/10.1016/j.datak.2021.101907.
URL https://www.sciencedirect.com/science/article/pii/S0169023X21000343

[36] M. Alenezi, H. A. Basit, F. I. Khan, M. A. Beg, A Comparison Study of Available Sofware Security Ontologies, in: Proceedings of the Evaluation and Assessment in Software Engineering, ACM, New York, NY, USA, 2020, pp. 499–504. doi:10.1145/3383219.3383292.
URL https://dl.acm.org/doi/10.1145/3383219.3383292

[37] Z. S. Ageed, R. K. Ibrahim, M. A. M. Sadeeq, Unified Ontology Implementation of Cloud Computing for Distributed Systems, Current Journal of Applied Science and Technology (2020) 82–97doi:10.9734/cjast/2020/v39i3431039.
URL https://journalcjast.com/index.php/CJAST/article/view/2963

[38] R. Penteado, M. Barreto, R. T. de Sousa Júnior, Security ontologies for cloud computing: a systematic literature review, Information Systems Frontiers 19 (2017) 735–758.

[39] B. A. Mozzaquatro, R. Jardim-Goncalves, C. Agostinho, Towards a reference ontology for security in the Internet of Things, in: 2015 IEEE International Workshop on Measurements & Networking (M&N), IEEE, 2015, pp. 1–6. doi:10.1109/IWMN.2015.7322984.
URL http://ieeexplore.ieee.org/document/7322984/

[40] P. Gonzalez-Gil, A. F. Skarmeta, J. A. Martinez, Towards an ontology for iot context-based security evaluation, in: 2019 Global IoT Summit (GIoTS), 2019, pp. 1–6. doi:10.1109/GIOTS.2019.8766400.

[41] V. M. Álvarez, J. M. Murillo, A. F. Gómez-Skarmeta, A semantic approach for managing security policies in the internet of things, IEEE Internet of Things Journal 6 (2019) 2781–2792.

[42] M. Alenezi, An Ontological Framework for Healthcare Web Applications Security, International Journal of Advanced Computer Science and Applications 12 (6) (2021) 511–516. doi:10.14569/IJACSA.2021.0120658.
URL http://thesai.org/Publications/ViewPaper?Volume=12&Issue=6&Code=IJACSA&SerialNo=58

[43] A. Nazir, S. Sholla, A. Bashir, An ontology based approach for context-aware security in the internet of things (iot), International Journal of Wireless and Microwave Technologies (IJWMT) 11 (1) (2021) 28–46.

[44] F. Alsubaei, A. Abuhussein, S. Shiva, Ontology-based security recommendation for the internet of medical things, IEEE Access 7 (2019) 48948–48960. doi:10.1109/ACCESS.2019.2910087.

[45] H. Belani, P. Solic, T. Perkovic, Towards Ontology-Based Requirements Engineering for IoT-Supported Well-Being, Aging and Health, in: 2022 IEEE 30th International Requirements Engineering Conference Workshops (REW), IEEE, 2022, pp. 65–74. arXiv:2211.10735, doi:10.1109/REW56159.2022.00019.
URL https://ieeexplore.ieee.org/document/9920130/

[46] F. Belala, A. Khamadkar, K. Motwani, An ontology-based approach for modeling security requirements of smart grid systems, Journal of Network and Computer Applications 91 (2017) 59–74.

[47] T. Qamar, N. Bawany, A Cyber Security Ontology for Smart City, International Journal on Information Technologies and Security 12 (3) (2020) 63–74.

[48] Y. I. Khan, M. U. Ndubuaku, Ontology-based automation of security guidelines for smart homes, in: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), 2018, pp. 35–40. doi:10.1109/WF-IoT.2018.8355214.

[49] C. Sánchez-Zas, V. A. Villagrá, M. Vega-Barbas, X. Larriva-Novo, J. I. Moreno, J. Berrocal, Ontology-based approach to real-time risk management and cyber-situational awareness, Future

Generation Computer Systems 141 (2023) 462–472. doi:https://doi.org/10.1016/j.future.2022.12.006.
URL https://www.sciencedirect.com/science/article/pii/S0167739X22004058

[50] J. Alanen, J. Linnosmaa, T. Malm, N. Papakonstantinou, T. Ahonen, E. Heikkilä, R. Tiusanen, Hybrid ontology for safety, security, and dependability risk assessments and Security Threat Analysis (STA) method for industrial control systems, Reliability Engineering & System Safety 220 (2022) 108270. doi:10.1016/j.ress.2021.108270.
URL https://linkinghub.elsevier.com/retrieve/pii/S0951832021007444

[51] I. Meriah, L. B. A. Rabai, Analysing Information Security Risk Ontologies, International Journal of Systems and Software Security and Protection 11 (1) (2020) 1–16. doi:10.4018/IJSSSP.2020010101.
URL https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/IJSSSP.2020010101

[52] Y. Merah, T. Kenaza, Ontology-based Cyber Risk Monitoring Using Cyber Threat Intelligence, ACM International Conference Proceeding Seriesdoi:10.1145/3465481.3470024.
URL https://dl.acm.org/doi/10.1145/3465481.3470024

[53] G. Engelberg, M. Fumagalli, A. Kuboszek, D. Klein, P. Soffer, G. Guizzardi, An ontology-driven approach for process-aware risk propagation, in: Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, SAC '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 1742–1745. doi:10.1145/3555776.3577795.
URL https://doi.org/10.1145/3555776.3577795

[54] R. Syed, Cybersecurity vulnerability management: A conceptual ontology and cyber intelligence alert system, Information & Management 57 (6) (2020) 103334. doi:https://doi.org/10.1016/j.im.2020.103334.

[55] G. Bakirtzis, T. Sherburne, S. Adams, B. M. Horowitz, P. A. Beling, C. H. Fleming, An ontological metamodel for cyber-physical system safety, security, and resilience coengineering, Software and Systems Modeling 21 (1) (2022) 113–137. arXiv:2006.05304, doi:10.1007/s10270-021-00892-z.
URL https://link.springer.com/10.1007/s10270-021-00892-z

[56] P. Yermalovich, M. Mejri, Ontology-based Model for Security Assessment: Predicting Cyberattacks through Threat Activity Analysis, International Journal of Network Security & Its Applications 12 (3) (2020) 1–22. doi:10.5121/ijnsa.2020.12301.
URL https://aircconline.com/ijnsa/V12N3/12320ijnsa01.pdf

[57] B. B. Duarte, R. de Almeida Falbo, G. Guizzardi, R. Guizzardi, V. E. S. Souza, An ontological analysis of software system anomalies and their

associated risks, Data and Knowledge Engineering 134, cited by: 8; All Open Access, Green Open Access. doi:10.1016/j.datak.2021.101892.
URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-85107656313&doi=10.1016%2fj.datak.2021.101892&partnerID=40&md5=6e482d2253dd4e42a10c098d9c7857ac

[58] R. Y. Venkata, P. Kamongi, K. Kavi, An ontology-driven framework for security and resiliency in cyber physical systems, in: 2018 The Thirteenth International Conference on Software Engineering Advances (ICSEA), 2018, pp. 13–19.

[59] M. Balduccini, E. Griffor, M. Huth, C. Vishik, M. Burns, D. Wollman, Ontology-based Reasoning about the Trustworthiness of Cyber-physical Systems, Living in the Internet of Things: Cybersecurity of the IoT - 2018 2018 (CP740) (2018) 12 (10 pp.)–12 (10 pp.). arXiv:1803.07438, doi:10.1049/cp.2018.0012.
URL https://digital-library.theiet.org/content/conferences/10.1049/cp.2018.0012

[60] M. C. Suárez-Figueroa, A. Gómez-Pérez, M. Fernández-López, The NeOn Methodology for Ontology Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 9–34. doi:10.1007/978-3-642-24794-1_2.
URL https://doi.org/10.1007/978-3-642-24794-1_2